ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Algorithms for Radio Networks

Localization

University of FreiburgTechnical Faculty
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Trilateration

- **Assuming the distance to three points is given**
- **System of equations**
  - **(x_i, y_i): coordinates of an anchor point i,**
  - **r distance from the anchor point i**
  - **(x_u, y_u): unknown coordinates of a node**

$$(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \text{ for } i = 1, \overset{2}{\ldots}, 3$$

- **Problem: Quadratic equations**
  - **Transformations lead to a linear system of equations**

Trilateration

- **System of equations**

$$(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \text{ for } i = 1, \ldots, 3$$

- **Transformation**

$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = r_1^2 - r_3^2$$
$$(x_2 - x_u)^2 - (x_2 - x_u)^2 + (y_2 - y_u)^2 - (y_2 - y_u)^2 = r_2^2 - r_3^2.$$

  - **results in:**

$$2(x_3 - x_1)x_u + 2(y_3 - y_1)y_u = (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$
$$2(x_3 - x_2)x_u + 2(y_3 - y_2)y_u = (r_2^2 - r_2^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)$$

Trilateration as a Linear System of Equations

- **Forming a system of equations**

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (r_2^2 - r_2^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix}$$

- **Example:**
  - **$(x_1, y_1) = (2,1)$, $(x_2, y_2) = (5,4)$, $(x_3, y_3) = (8,2)$,**
  - **$r_1 = 10^{1/2}$, $r_2 = 2$, $r_3 = 3$**

$$2\begin{bmatrix} 6 & 1 \\ 3 & -2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} 64 \\ 22 \end{bmatrix}$$

➔ $(x_u,y_u) = (5,2)$

Trilateration as a Linear System of Equations

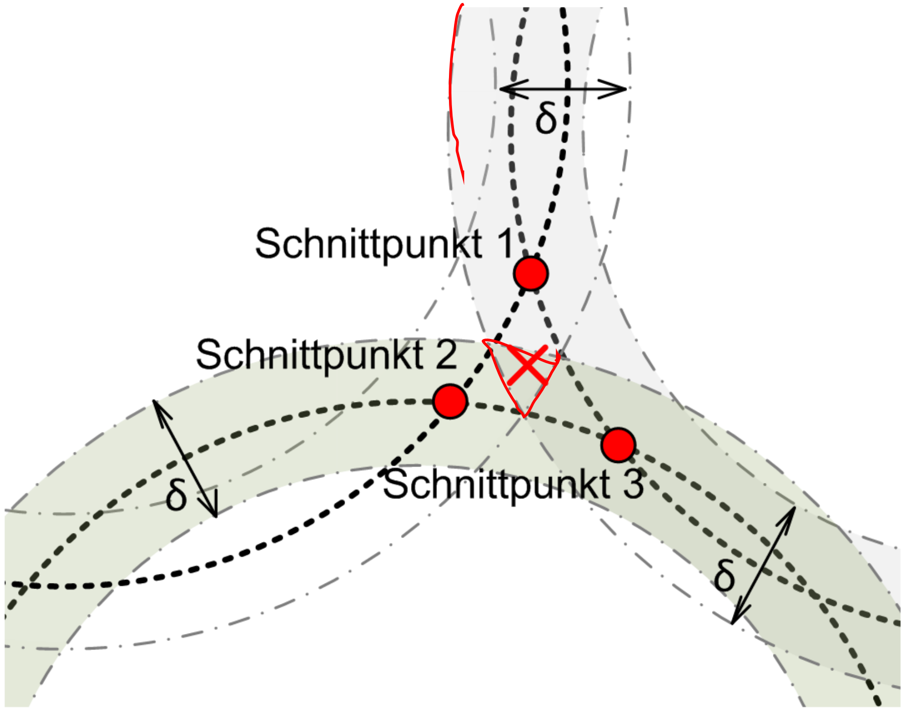
- **In three dimensions**
  - **Intersection of four spheres**

$$\underbrace{\begin{bmatrix} (d_1^2-d_4^2)-(x_1^2-x_4^2)-(y_1^2-y_4^2)-(z_1^2-z_4^2) \\ (d_2^2-d_4^2)-(x_2^2-x_4^2)-(y_2^2-y_4^2)-(z_2^2-z_4^2) \\ (d_3^2-d_4^2)-(x_3^2-x_4^2)-(y_3^2-y_4^2)-(z_3^2-z_4^2) \end{bmatrix}}_{\vec{b}} = 2 \underbrace{\begin{bmatrix} (x_4-x_1)(y_4-y_1)(z_4-z_1) \\ (x_4-x_2)(y_4-y_2)(z_4-z_2) \\ (x_4-x_3)(y_4-y_3)(z_4-z_3) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_{P1} \\ y_{P1} \\ z_{P1} \end{bmatrix}}_{\vec{x}}$$

- **Solve** $\mathbf{Ax = b} \Rightarrow \mathbf{x = A^{-1}b}$

Trilateration

- **In case of measurement errors**

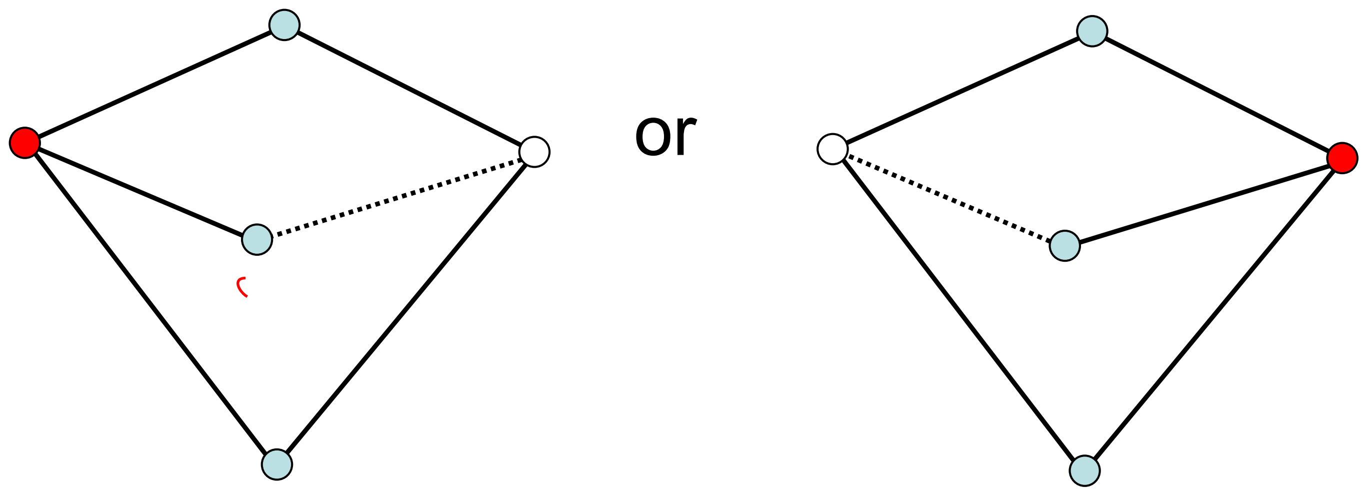


[F. Höflinger, 2013]

- **Averaging: e.g. centroid of triangle**

Trilateration

- **Measurement errors**
  - **Small distance errors can lead to large position errors**



- **flip ambiguity from noise**

Multilateration with *absolute* distances

- Multilateration (absolute distances): Calculate the intersection of *at least four* distance measurements
  - May be over-determined equation system: More equations than variables
  - "No solution" in case of measurement errors
- Minimize sum of quadratic residuals: *Least squares*
- Vector notation
  - Solve $(\mathbf{A}^T\mathbf{A})\mathbf{x} = \mathbf{A}^T\mathbf{b} \Rightarrow \mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\, \mathbf{A}^T\mathbf{b}$
  - Matrix inverse by Gauss-Jordan elimination

Multilateration with *relative* distances

- **Multilateration (relative): Calculate the intersection of *relative* distance measurements**
  - **Emission time e unknown!**
  - **Measure only reception times T_i, i = 1, ..., n**
  - **System of equations $T_i = e + \| r_i - s \| / c$**
  - **...for a signal traveling from s to receivers r_i**
- **Subtract two absolute times T_i and T_j:**
  - $T_i - T_j = \| r_i - s \| / c - \| r_j - s \| / c =: \Delta t \quad (i, j = 1, ..., n)$
  - **System of hyperbolic equations**
  - **Time Difference of Arrival Δt, relative distance $\Delta d = c \Delta t$**

Multilateration with *relative* distances

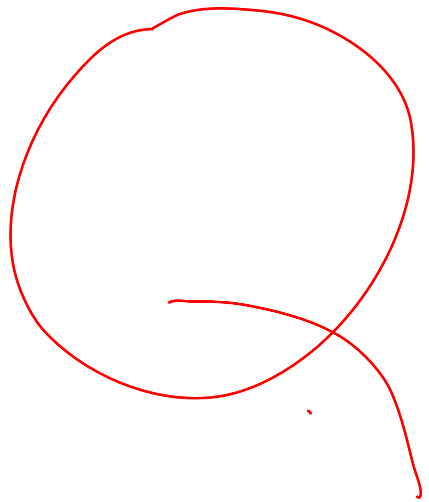
- Advantages
  - No cooperation of signal emitter
  - Hardware delays cancel out (both emitter and receiver)
  - Passive localization / natural signal sources
- Disadvantages
  - Larger number of unknown values: Position and time
  - Synchronization still (usually) required

Anchor-free localization

- **"Anchor-free localization":**
  - **Hyperbolic multilateration**
  - **Unknown emitters $\mathbf{s}_j$, and unknown receivers $\mathbf{r}_i$**
- **Advantages:**
  - **No need to measure receiver positions**
  - **Self-positioning by passive information from the surroundings**
- **Disadvantages:**
  - **Even larger number of unknown variables**

Anchor-free localization

- **For absolute distances** d_{ik}**:**
  - **Solve** $\| \mathbf{r}_i - \mathbf{s}_k \| = d_{ik} \qquad (i, j = 1, \ldots, n\ ;\ \ k = 1, \ldots, m)$
  - **Problem of intersecting circles / spheres**
  - **Bipartite distance graph:** $G = \left(\{\mathbf{r}_i\},\ \{\mathbf{s}_k\},\ \{d(i, k)\} \right)$
  - **Minimum case closed-from solutions known** [Kuang, et al., ICASSP 2013]



Anchor-free localization

- For ***relative*** distances $\Delta d_{ijk} = d_{ik} - d_{jk}$:
  - Solve $\| \mathbf{r}_i - \mathbf{s}_k \| - \| \mathbf{r}_j - \mathbf{s}_k \| = \Delta d_{ijk}$
  - Problem of intersecting hyperbolas / hyperboloids
  - Closed-form solutions only for larger problem sets
    [Pollefeys and Nister, ICASSP 2008], [Kuang and Åström, EUSIPCO 2013]
  - Minimum problem set: Iterative/recursive approximations
    [Wendeberg and Schindelhauer, Algosensors 2012]

Anchor-free localization

- **Degrees of freedom**

$$T_{ik} = e_{ik} + \| \mathbf{r}_i - \mathbf{s}_k \| / \mathbf{c}$$

(e_{ik}, $\mathbf{r}_i$, $\mathbf{s}_k$ unknown)

signal sources	receivers 1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	3	3	3	3	3	3	3	3
3	5	4	3	2	1	0	-1	-2
4	7	5	3	1	-1	-3	-5	-7
5	9	6	3	0	-3	-6	-9	-12
6	11	7	3	-1	-5	-9	-13	-17
7	13	8	3	-2	-7	-12	-17	-22
8	15	9	3	-3	-9	-15	-21	-27
9	17	10	3	-4	-11	-18	-25	-32
10	19	11	3	-5	-13	-21	-29	-37
11	21	12	3	-6	-15	-24	-33	-42
12	23	13	3	-7	-17	-27	-37	-47

$$G_{2D} = 2n + 3m - nm - 3$$

signal sources	receivers 1	2	3	4	5	6	7	8
1	0	2	4	6	8	10	12	14
2	3	4	5	6	7	8	9	10
3	6	6	6	6	6	6	6	6
4	9	8	7	6	5	4	3	2
5	12	10	8	6	4	2	0	-2
6	15	12	9	6	3	0	-3	-6
7	18	14	10	6	2	-2	-6	-10
8	21	16	11	6	1	-4	-9	-14
9	24	18	12	6	0	-6	-12	-18
10	27	20	13	6	-1	-8	-15	-22
11	30	22	14	6	-2	-10	-18	-26
12	33	24	15	6	-3	-12	-21	-30

$$G_{3D} = 3n + 4m - nm - 6$$

Anchor-free localization

- **Minimum cases**

	2D	3D
general setting	**4 / 6**	**5 / 10** **6 / 7**
far-field setting	**3 / 3 (sync.)** **3 / 5 (unsync.)**	**4 / 6 (sync.)** **4 / 9 (unsync.)**

Minimum number of required receivers / emitters

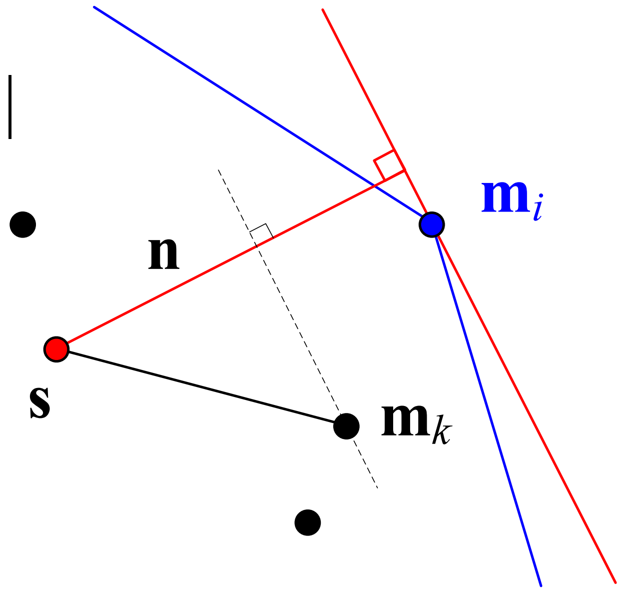
Anchor-free localization

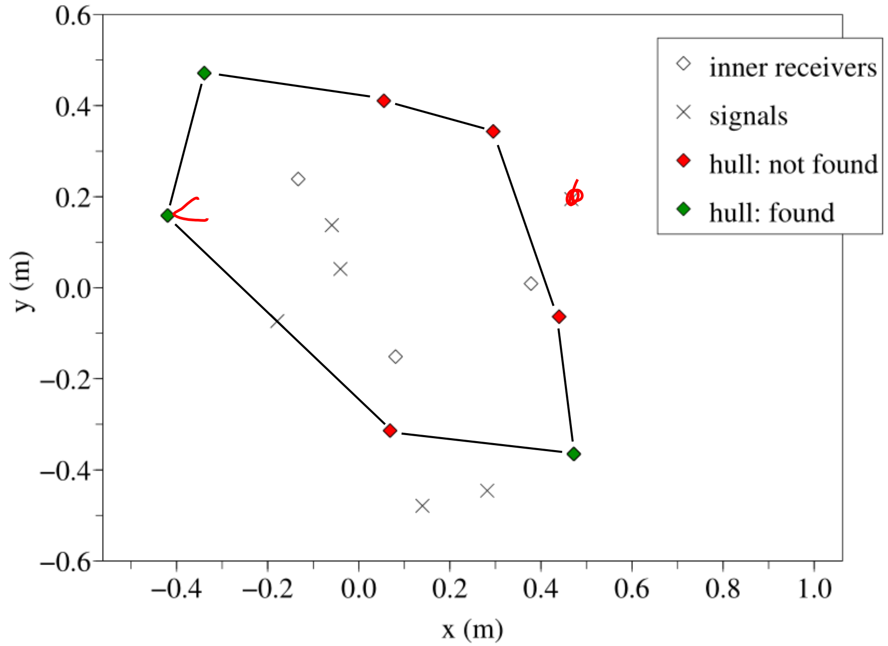
- **Strategies:**
  - **(1.) Estimate receiver topology from known information**
  - **(2.) Assume large number of emitters and receivers**
  - **(3.) Assume specific distribution of emitters and receivers**
  - **(4.) Heat the CPU: Optimization, branch-and-bound search, ...**

Anchor-free localization

- (1.) Topology: Hull element
  - "The receiver which receives the last timestamp is an element of the convex hull"

$$\mathbf{n}_0 = \mathbf{n} \,/\, \|\mathbf{n}\|$$



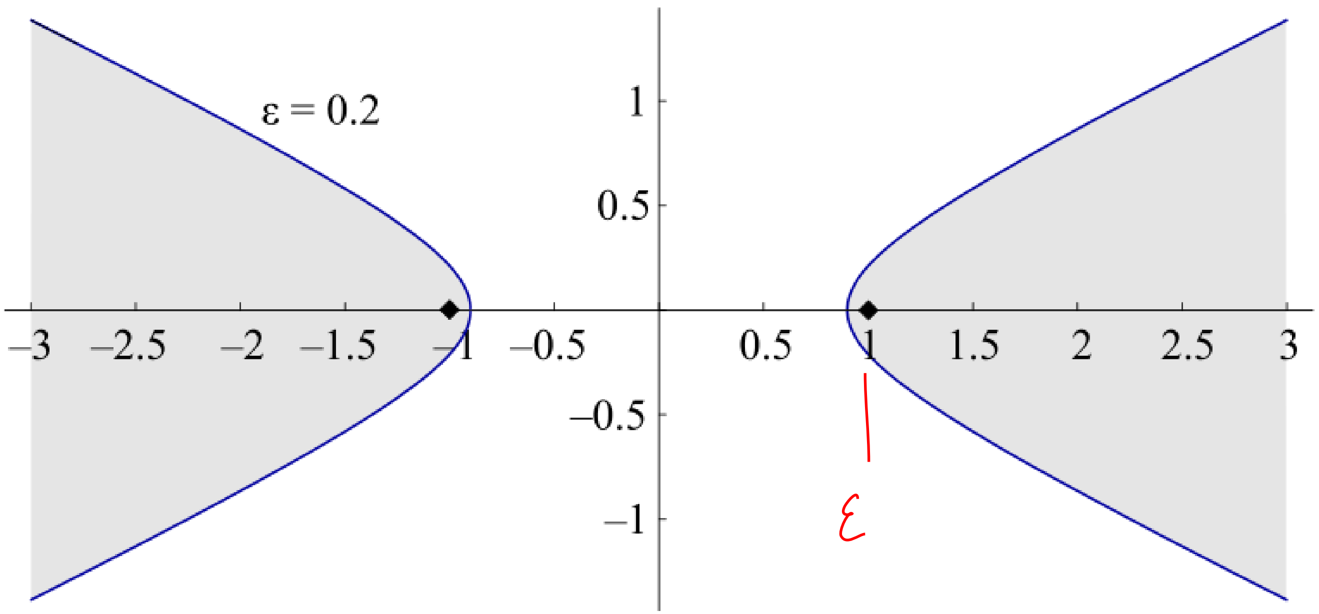


If exists i such that for all k: $T_i \geq T_k$, then holds:

$$(\mathbf{m}_i - \mathbf{s})^T\, \mathbf{n}_0 = \|\mathbf{m}_i - \mathbf{s}\| \geq \|\mathbf{m}_k - \mathbf{s}\| \geq (\mathbf{m}_i - \mathbf{s})^T\, \mathbf{n}_0$$

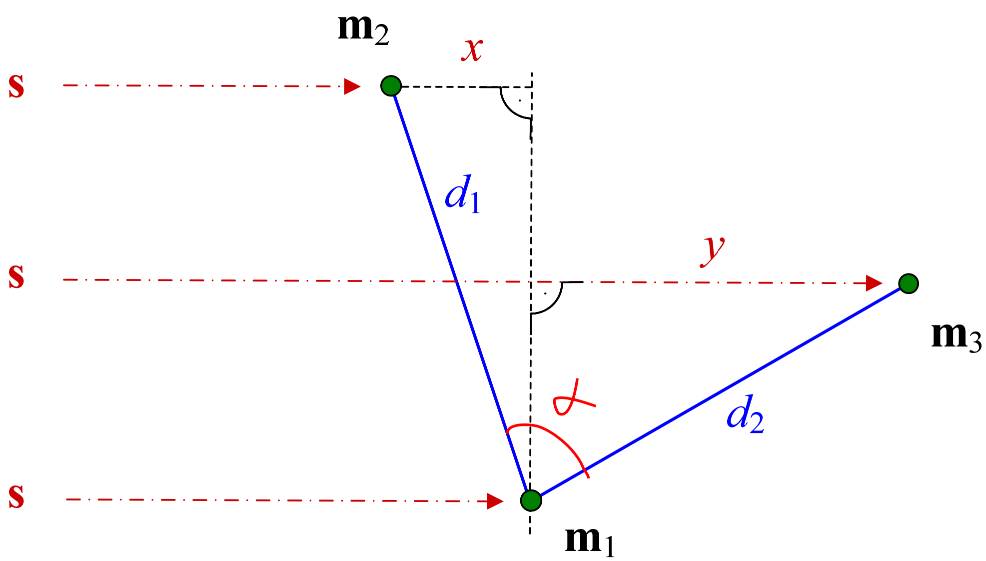
Anchor-free localization

- **(2.) Large number of signals: Statistical assumptions**
  [Schindelhauer, et al., SIROCCO 2011]
  - **Lemma: Many signals occur from the long side of any two receivers.**
  - **Estimate the distance:** $d \sim c/2\,(\Delta t_{max} - \Delta t_{min})$

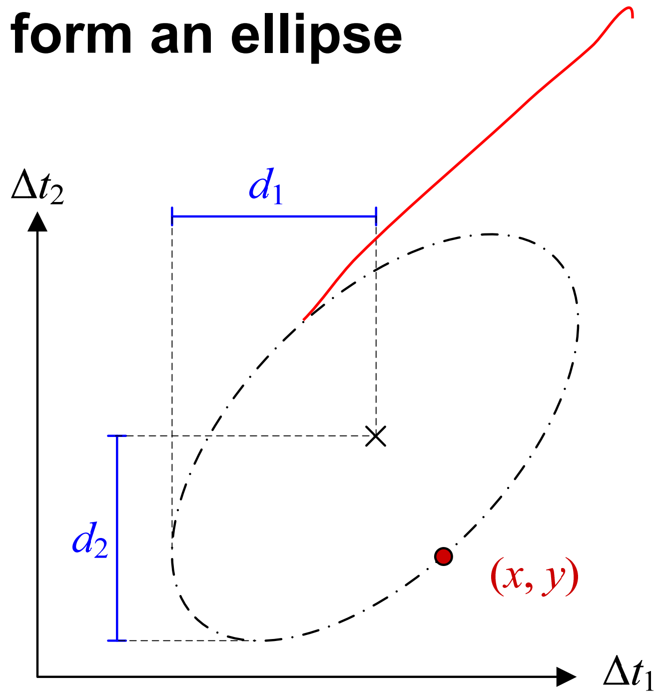


Anchor-free localization

- **(3.) Assume that signals occur from far away:**
  - **“far-field assumption”, linear frontier of signal propagation**
- **The Ellipsoid TDoA Method** [Wendeberg, et al., TCS, 2012]
  - **Time differences of *three* receivers form an ellipse**



top-down view



time differences

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithms for Radio Networks

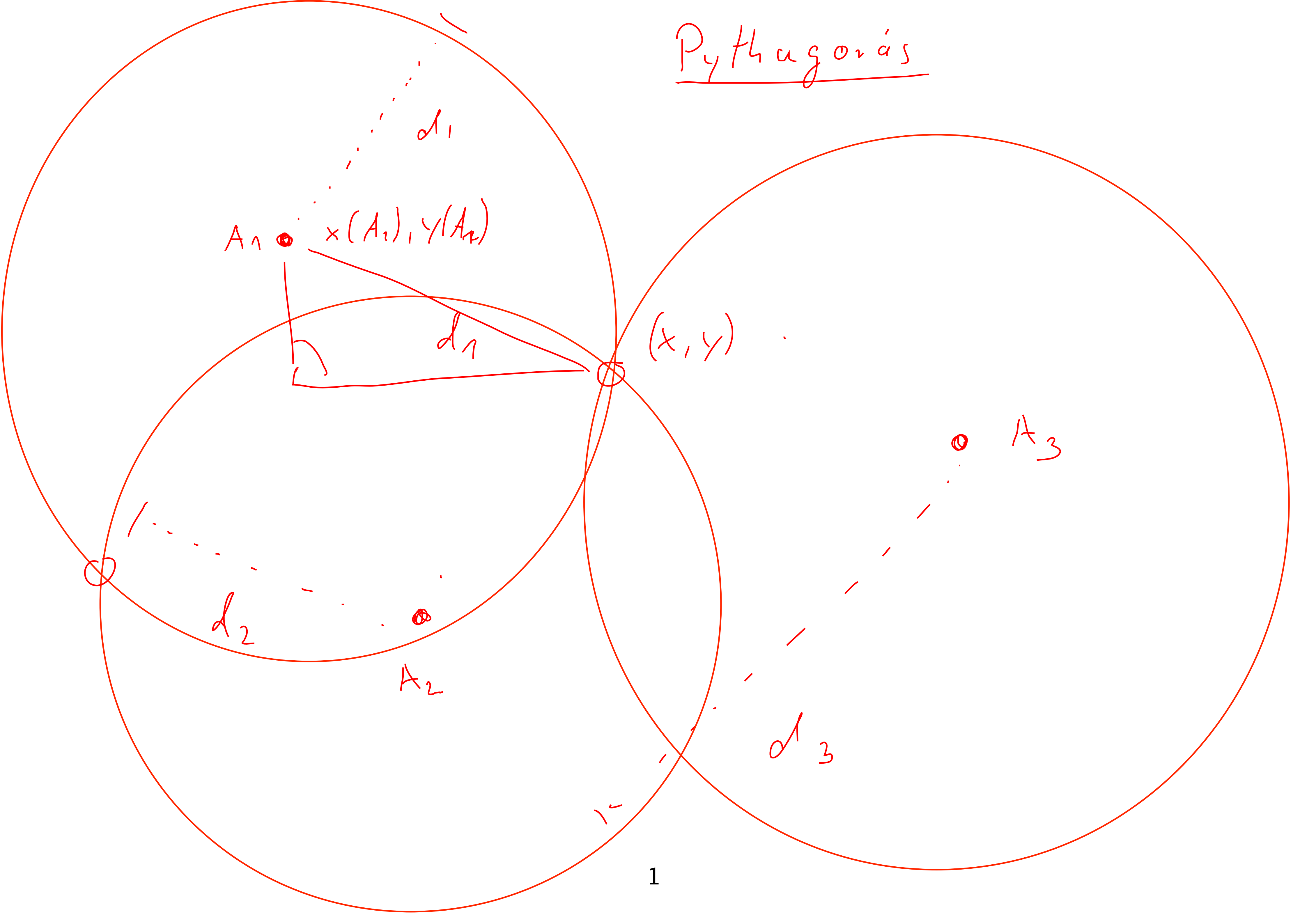
Localization

University of FreiburgTechnical Faculty
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

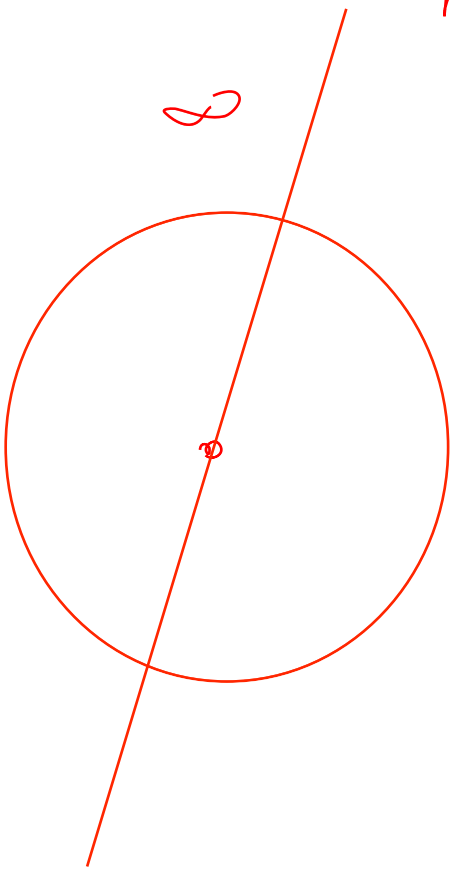
Pythagoras

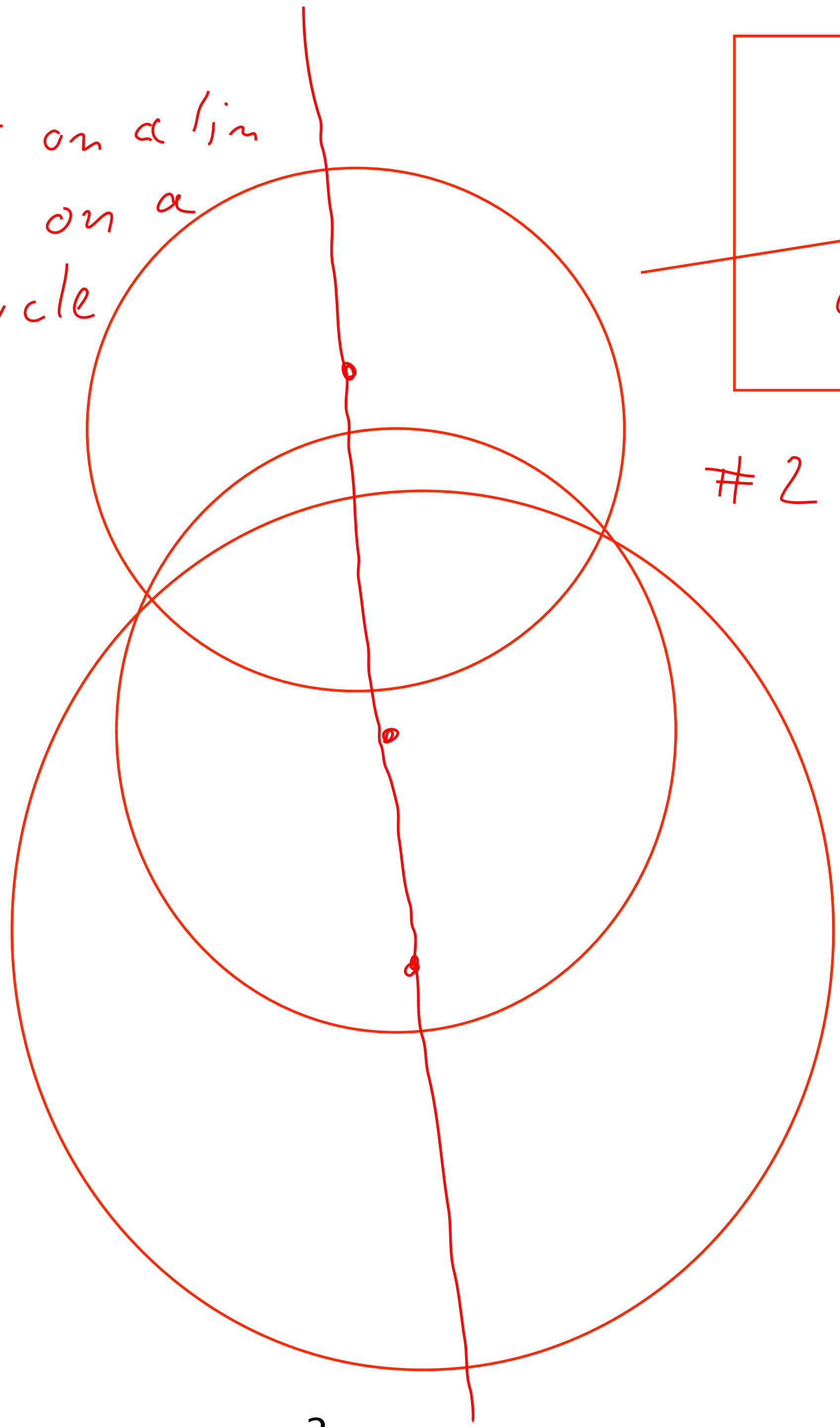


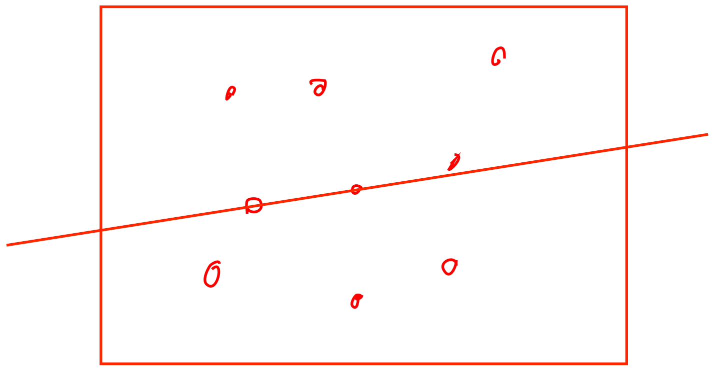
general position

- no 3 points on a line
- no 4 point on a circle

∞







#2

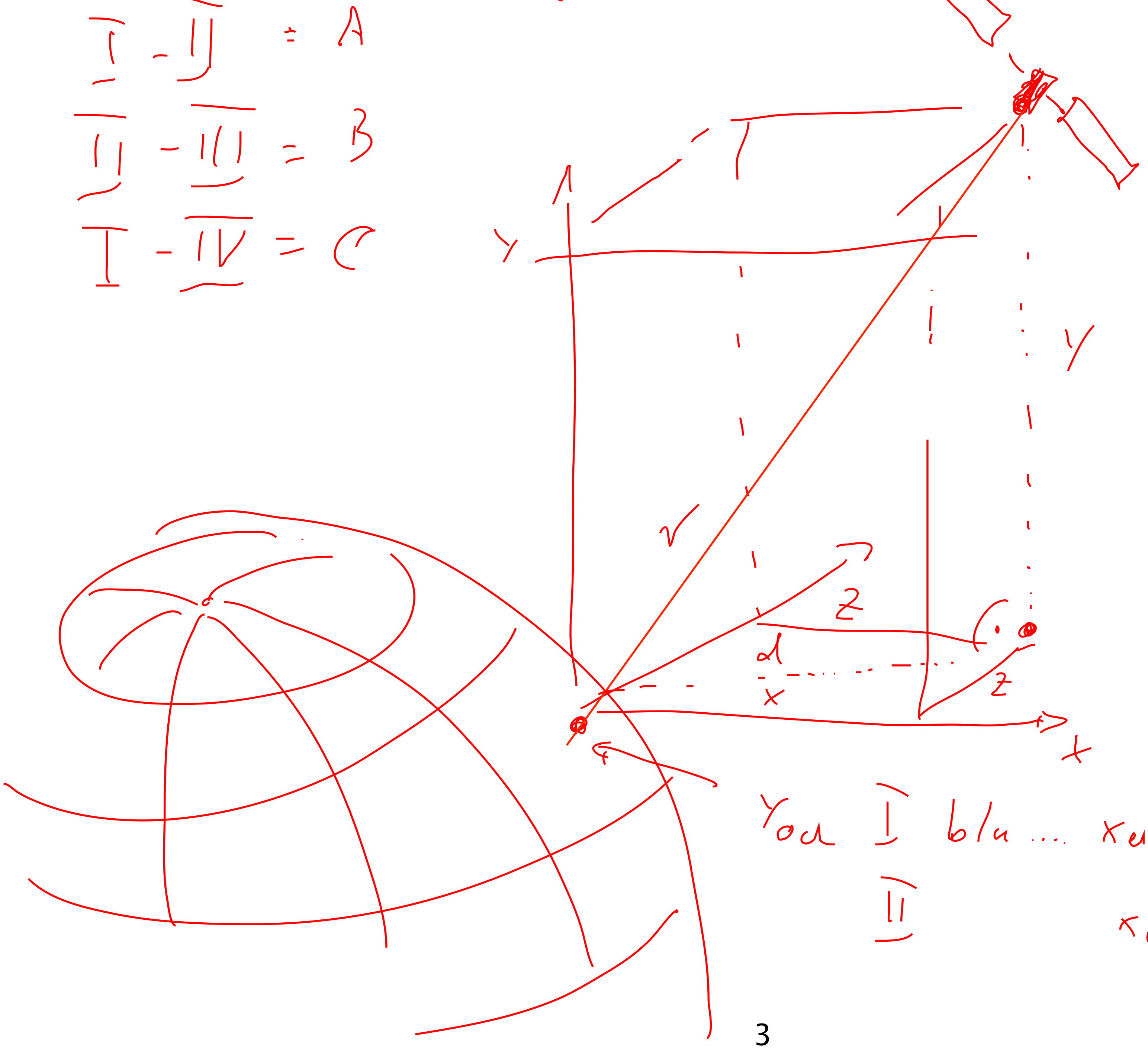
$$\frac{0}{>0} = 0$$

$$C = A + B$$

$$\overline{\underline{\mathrm{I}}} - \overline{\underline{\mathrm{II}}} = A$$

$$\overline{\underline{\mathrm{II}}} - \overline{\underline{\mathrm{III}}} = B$$

$$\overline{\underline{\mathrm{I}}} - \overline{\underline{\mathrm{IV}}} = C$$



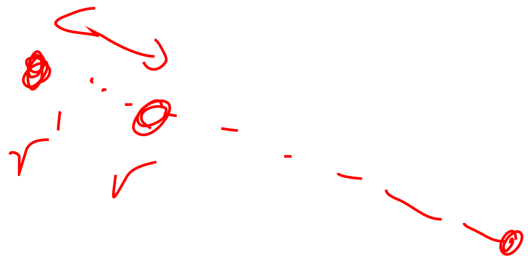
$$d^2 = x^2 + z^2$$

$$d^2 + y^2 = r^2$$

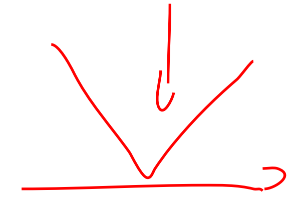
$$x^2 + y^2 + z^2 = r^2$$

You I bla .... x_u^2 ... y_u^2 ... z_u^2 ... = ...

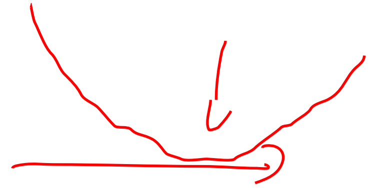
II x_u^2 ... y_u^2 ... z_u^2 ...



$$|r - r'|$$



$$(r - r')^2$$

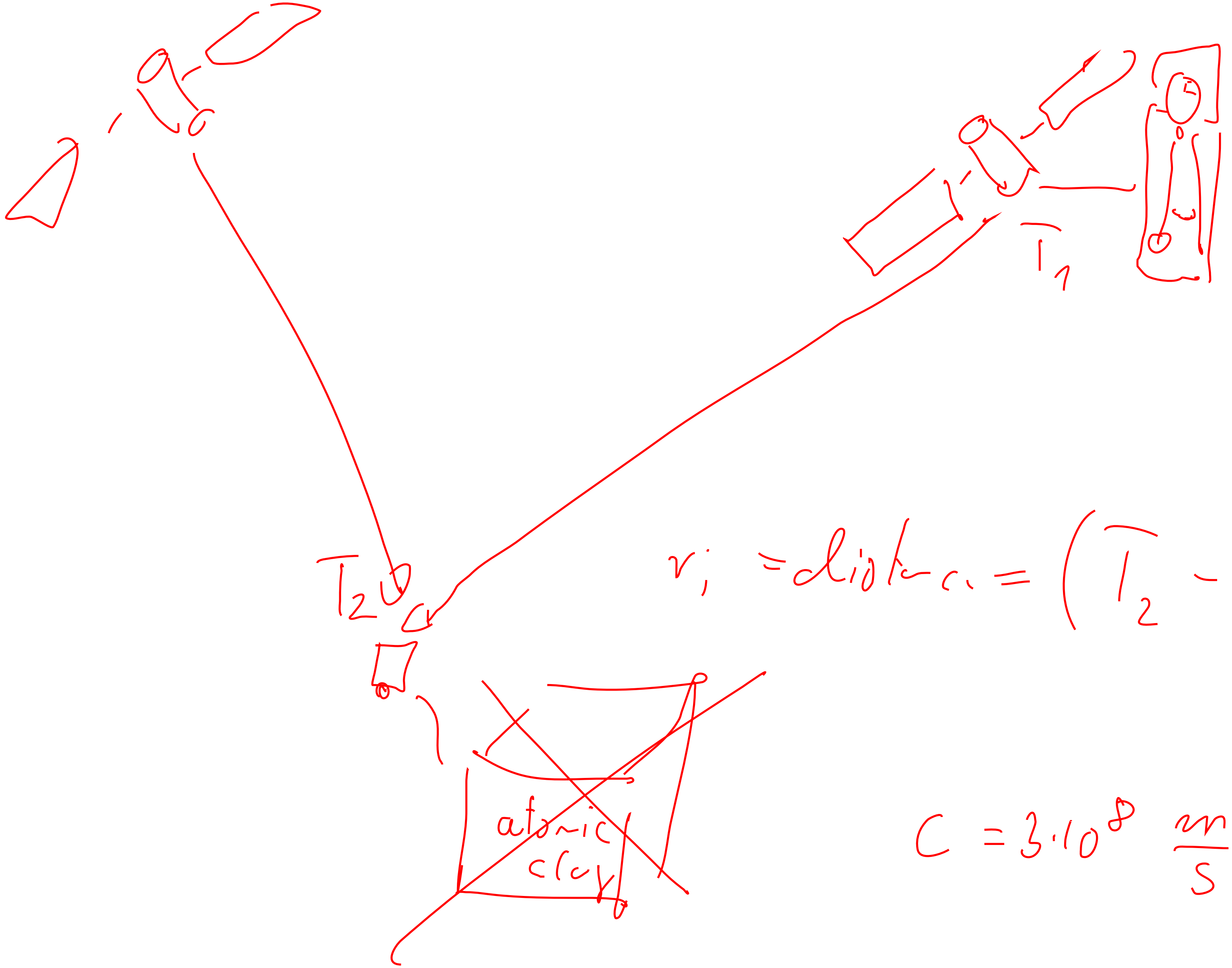


$$\frac{d(x - r')^2}{dx} = 2x - 2r'$$

$$A^T \overset{\shortparallel}{}$$

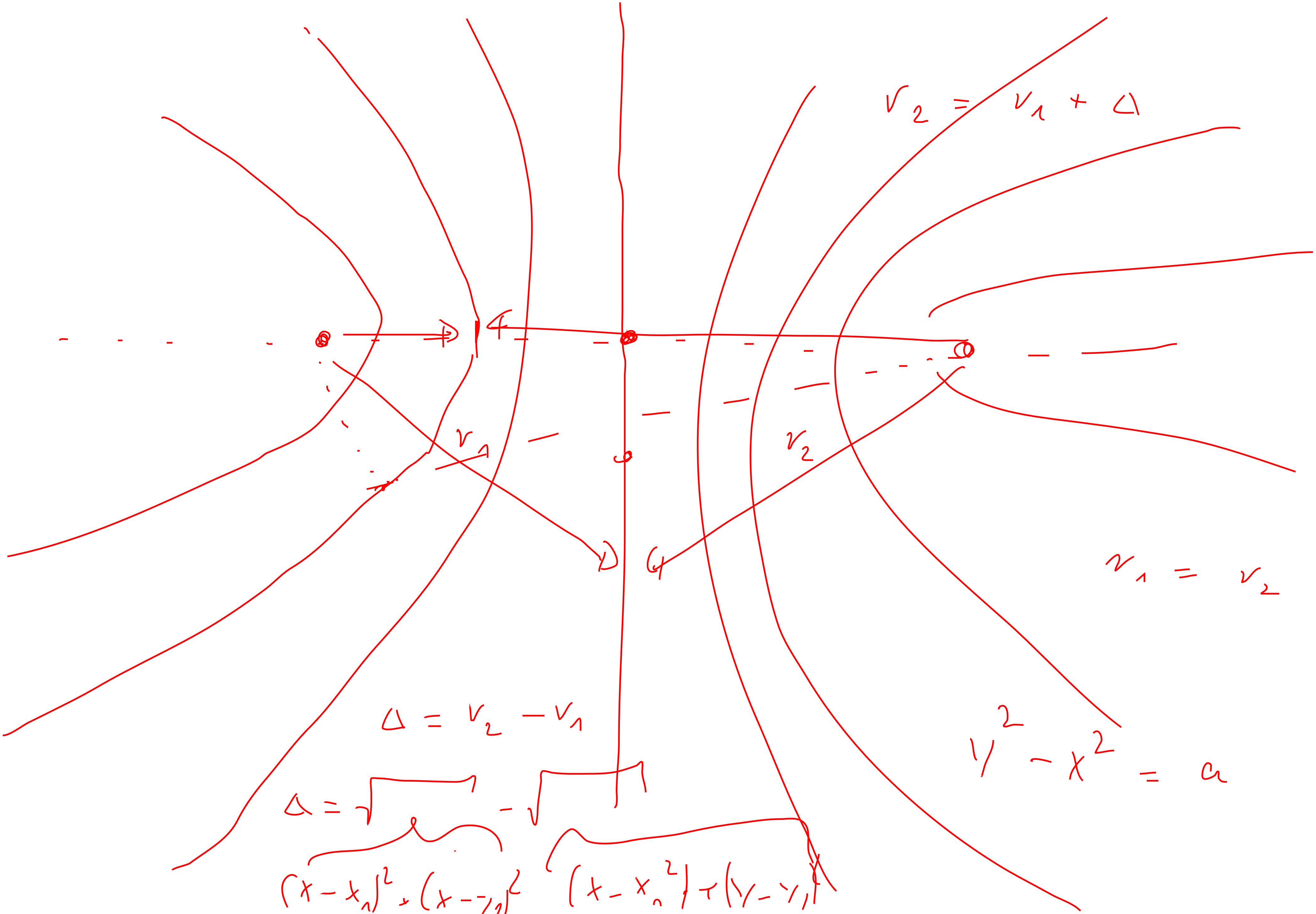
$$\underbrace{\begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 2 & 4 & 6 & 8 & 10 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \\ 9 & 10 \end{pmatrix}} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = A^T \begin{pmatrix} v_1 \\ v_2 \\ v_4 \\ v_5 \\ v_3 \end{pmatrix}$$

$$\begin{pmatrix} 211 & 1003 \\ [illegible] & [illegible] \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} [illegible] \\ [illegible] \end{pmatrix}$$



$$r_i = \text{distance} = (T_2 - T_1)\cdot c$$

$$c = 3\cdot 10^8 \frac{m}{s} = \frac{d}{t}$$



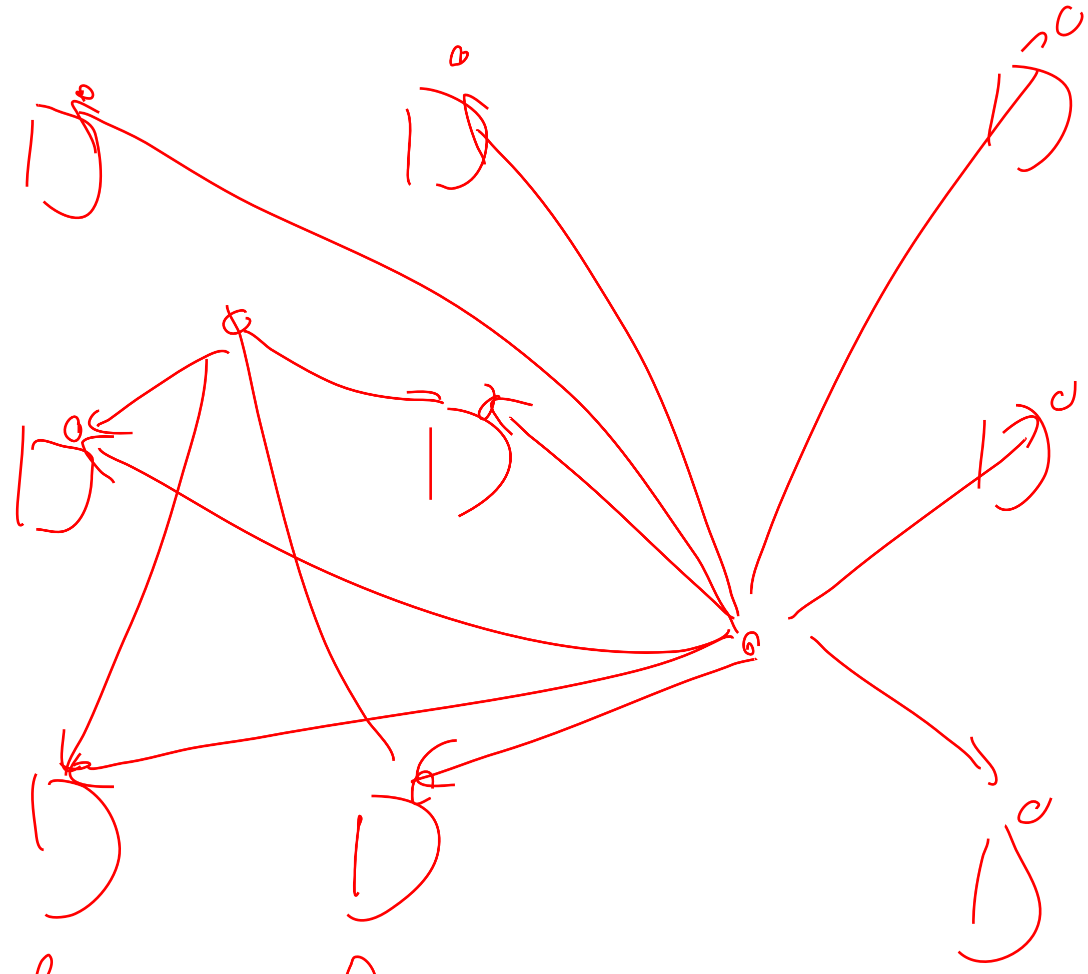
$$r_2 = r_1 + \Delta$$

$$r_1 = r_2$$

$$\Delta = r_2 - r_1$$

$$y^2 - x^2 = a$$

$$\Delta = \sqrt{\underbrace{(x-x_1)^2+(y-y_1)^2}} - \sqrt{\underbrace{(x-x_2)^2+(y-y_2)^2}}$$



n : # Sender

m : # Receiver.

$$2 \cdot n + 2 \cdot m + n$$

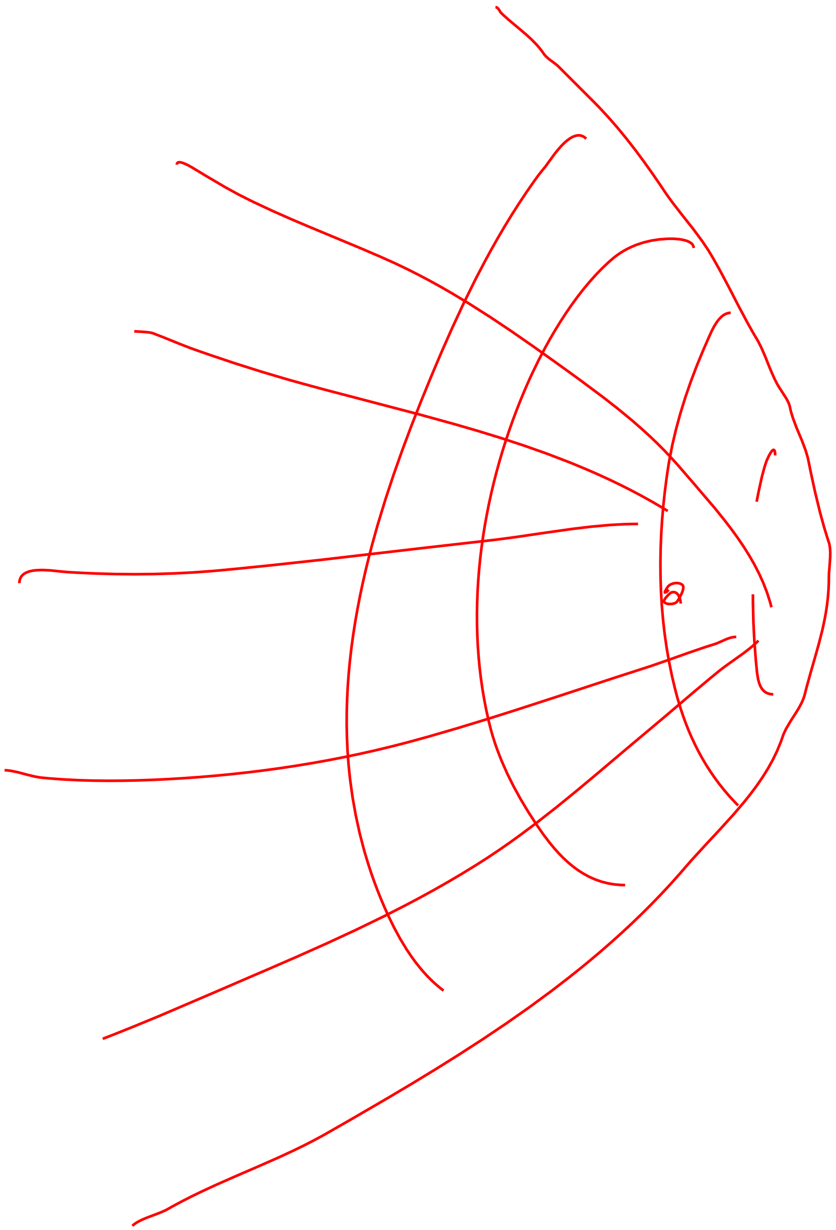
x/y   x/y   time

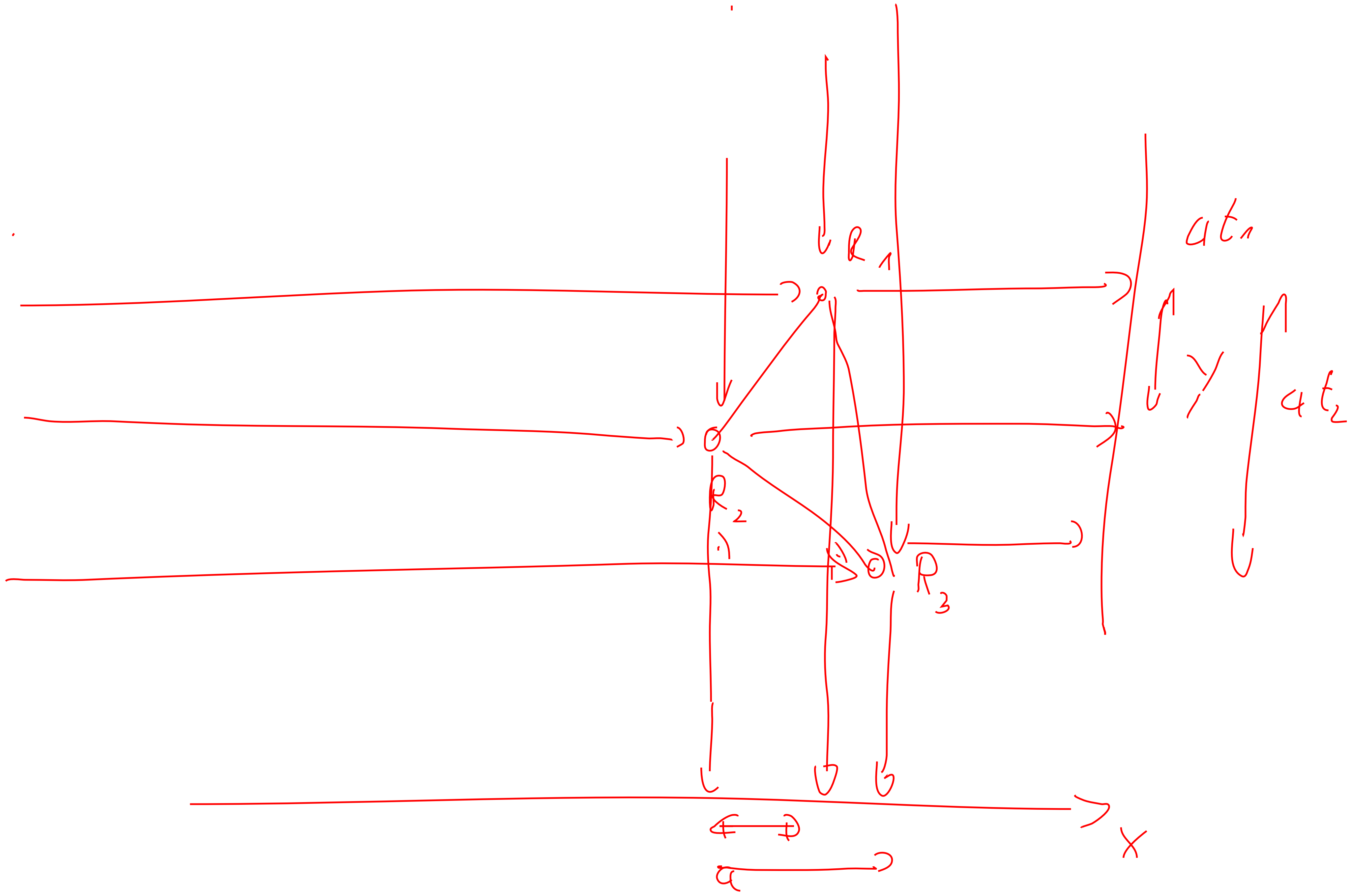
$$-2-1$$

---

equations : $n \cdot m$

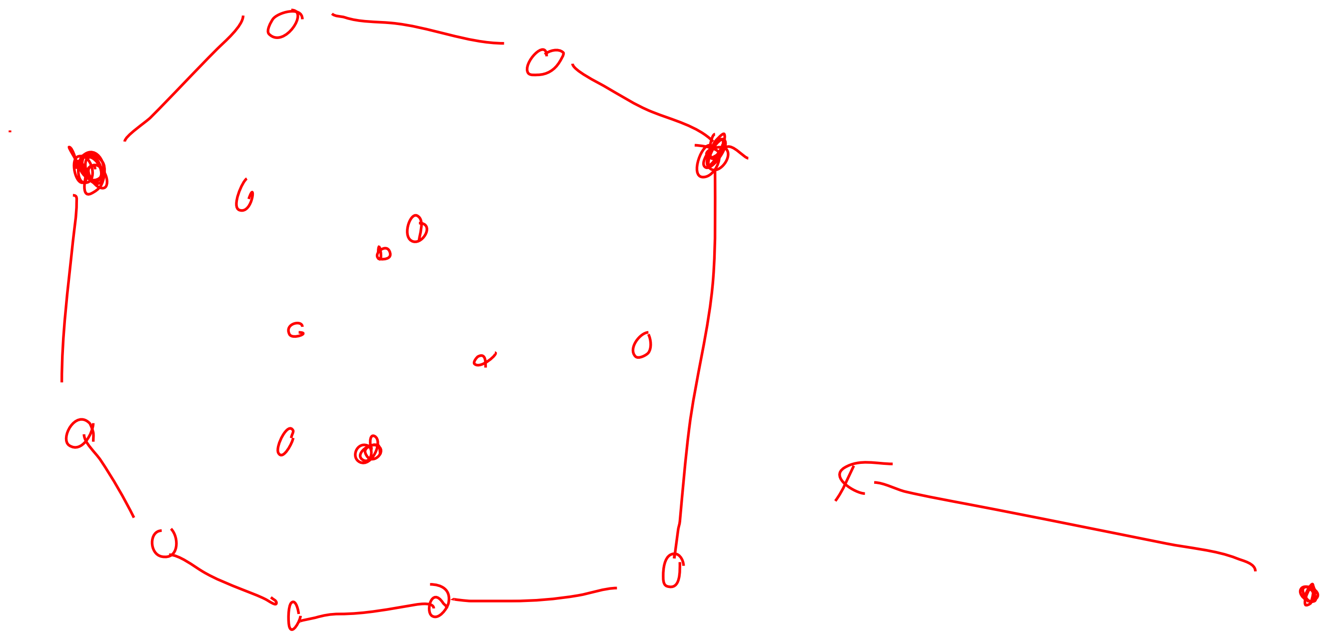
m

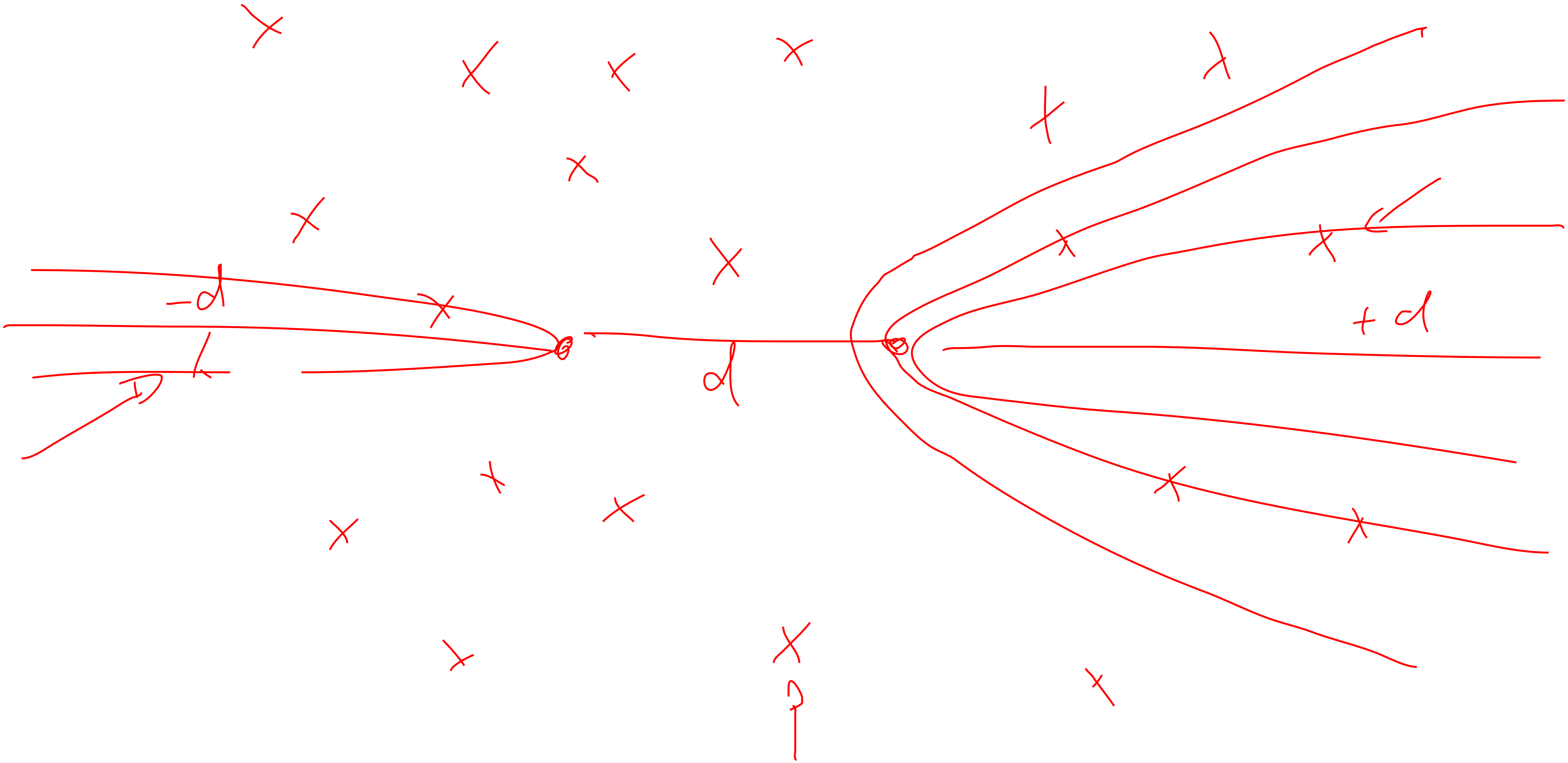




R_1
R_2
R_3
$a\,t_1$
$a\,t_2$
y
x
a

convex hull




-d
D
d
+d
j